



Global hydroelastic model for springing and whipping based on a free-surface CFD code (OpenFOAM)

Seng, Sopheak; Jensen, Jørgen Juncher; Malenica, Sime

Published in:
International Journal of Naval Architecture and Ocean Engineering

Link to article, DOI:
[10.2478/IJNAOE-2013-0229](https://doi.org/10.2478/IJNAOE-2013-0229)

Publication date:
2014

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Seng, S., Jensen, J. J., & Malenica, S. (2014). Global hydroelastic model for springing and whipping based on a free-surface CFD code (OpenFOAM). *International Journal of Naval Architecture and Ocean Engineering*, 6(4), 1024-1040. <https://doi.org/10.2478/IJNAOE-2013-0229>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Global hydroelastic model for springing and whipping based on a free-surface CFD code (OpenFOAM)

Sopheak Seng¹, Jørgen Juncher Jensen¹ and Šime Malenica²

¹*Department of Mechanical Engineering, Technical University of Denmark, DK 2800, Denmark*

²*Bureau Veritas, 92200 Neuilly-sur-Seine, France*

ABSTRACT: *The theoretical background and a numerical solution procedure for a time domain hydroelastic code are presented in this paper. The code combines a VOF-based free surface flow solver with a flexible body motion solver where the body linear elastic deformation is described by a modal superposition of dry mode shapes expressed in a local floating frame of reference. These mode shapes can be obtained from any finite element code. The floating frame undergoes a pseudo rigid-body motion which allows for a large rigid body translation and rotation and fully preserves the coupling with the local structural deformation. The formulation relies on the ability of the flow solver to provide the total fluid action on the body including e.g. the viscous forces, hydrostatic and hydrodynamic forces, slamming forces and the fluid damping. A numerical simulation of a flexible barge is provided and compared to experiments to show that the VOF-based flow solver has this ability and the code has the potential to predict the global hydroelastic responses accurately.*

KEY WORDS: Hydroelasticity; Fluid-structure interaction (FSI); Volume of fluid (VOF); CFD; OpenFOAM; Modal superposition.

INTRODUCTION

CFD simulations of free surface flows are gaining more attentions in marine, offshore and ship applications due to the possibility to predict loads and structural responses in realistic sea conditions. Recent publications (Kim et al., 2009; Cabos et al., 2011; Oberhagemann et al., 2012; Piro and Maki, 2013) have extended the capability of the simulations to account for hydroelastic effects as this capability is essential in order to investigate springing and slamming-induced whipping responses in ships. There are several aspects of the numerical implementation which needs to be addressed due to the complicated nature of the hydroelastic phenomena. One of them is the coupling between the structural solver and the flow solver. Although this coupling has been proved manageable (see e.g. Hou et al., 2012) it is still challenging to formulate an efficient and, at the same time, numerically stable Fluid-Structure Interaction (FSI) scheme. Another challenging aspect of the implementation is related to an energy conserving grid-to-grid mapping between the structural solver and the flow solver.

The present work aims for extending the capabilities of OpenFOAM (an open source CFD software package, Weller et al., 1998) to simulate springing and slamming-induced whipping responses on large vessels moving in waves in any heading. A theoretical description of the code is provided which covers:

Corresponding author: Sopheak Seng, e-mail: sose@mek.dtu.dk

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

- The VOF-based free surface flow solver formulated in an ALE (arbitrary Lagrangian-Eulerian) frame
- The flexible body motion solver where the local deformation is approximated by a modal superposition of dry mode shapes and the rigid body motion is solved for such that nonlinearity in the rigid body motion and its coupling with the local deformation is fully preserved
- A partitioned FSI (fluid-structure interaction) scheme with the Aitken's acceleration (Irons and Tuck, 1969) for a strongly couple FSI solution
- The transferring of the displacement and fluid forces between the flow and motion solver which in the present formulation allows the force distribution from the fluid solver to be transferred in the modal spaces; thus eliminating the need for a grid-to-grid mapping of the force distribution

FLOW SOLVER

The flow field inside the fluid domain, which contains both air and water, is governed by the solution of the incompressible Navier-Stokes equations with the free-surface captured by the volume of fluid method (VOF, Hirt and Nichols, 1981). The flow equations are written in an Arbitrary Lagrangian-Eulerian frame (ALE) as follows:

$$\nabla \cdot \mathbf{u} = 0 \quad (1)$$

$$\frac{\partial \alpha}{\partial t} + \mathbf{u}_c \cdot \nabla \alpha + \nabla \cdot [\alpha(1-\alpha)\mathbf{u}_r] = 0 \quad (2)$$

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u}_c^T) - \nabla \cdot [\mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)] = -\nabla p_d - (\mathbf{g} \cdot \mathbf{x}) \nabla \rho \quad (3)$$

where $\mathbf{u}, \alpha, p_d, \mathbf{g}, \mathbf{x}$ are the velocity field, the volume phase fraction field, the dynamic pressure field, the gravitational acceleration vector and the position vector, respectively. The effect of the surface tension has been neglected. The convective velocity \mathbf{u}_c is evaluated as $\mathbf{u}_c = \mathbf{u} - \mathbf{u}_m$ where \mathbf{u}_m is the domain velocity emerged from the ALE formulation. The pressure p_d is the pressure field without the hydrostatic pressure, defined as $p_d = p - \rho \mathbf{g} \cdot \mathbf{x}$ where p is the static pressure. The fluid properties ρ and μ are the effective density and dynamic viscosity for the air-water mixture defined as

$$\rho = \alpha \rho_w + (1 - \alpha) \rho_a$$

$$\mu = \alpha \mu_w + (1 - \alpha) \mu_a \quad (4)$$

where the subscript “w” and “a” indicate the properties of water and air, respectively. The volume phase fraction α is defined according to the VOF formulation as a bounded non-dimensional value between 0 and 1, where the value 0 indicates a control volume filled with air and 1 if the control volume is filled with water. At the air-water interface, α takes an intermediate value between these intervals; elsewhere its value shall be either 0 or 1 exactly. Eq. (2) is the transport equation for α with an artificial compressive term (the third term, see Rusche, 2002) added for the purpose to aid the numerical solution to maintain a sharp interface. With this artificial term, there is no need for special numerical treatments of the convective terms (the second term) e.g. HRIC (Muzaferija et al., 1999), CICSAM (Ubbink and Issa, 1999) or complicated interface reconstruction technique such as PLIC (Youngs, 1982). This term acts against the numerical diffusion at the vicinity of the interface with its compressive velocity field \mathbf{u}_r defined to have its direction normal to the instantaneous free surface. The magnitude of \mathbf{u}_r , however, is not well defined and has been determined empirically. The formulation of \mathbf{u}_r applied in this work is described accurately in Berberovic et al. (2009).

The equations are discretized using body-fitted unstructured finite volume cells with a collocated arrangement of the primitive variables. The velocity-pressure coupling is handled using a variant of the PISO-scheme (Issa, 1985) which incorporates iterations on the convective velocity field to achieve a better accuracy, stability, and to allow a larger time step. The original PISO-algorithm handles the velocity-pressure coupling by first predicting the momentum using the previously known convective velocity and pressure fields and second correcting the momentum at least two times through the solutions of the pressure equation. This is also known as the PISO-loop which is done once only in the original PISO-scheme by Issa 1985. The variant of the PISO-scheme implemented in the code repeats this PISO-loop several times, each time with the latest known convective velocity and pressure fields. In the process, the solution of the volume phase fraction is updated according to the latest known convective velocity.

Before solving for the volume phase fraction and the velocity-pressure coupling, a mesh update is executed following by an estimation of the mesh velocity and the effective convective velocity. The internal grid points can be moved virtually arbitrary, see e.g. Jasak and Tuković (2006). The only requirement here is that the resulting finite volume cells must have a sufficient quality to preserve the stability and the accuracy of the numerical solution. For the boundary points, it is required by the body boundary condition (slip or no-slip) that the boundary which represents the body shall be moved according to the dynamic motion and structural deformation of the body. Without the local structural deformation a mesh update can be done conveniently and efficiently by moving the whole mesh rigidly according to the rigid translational and rotational motion of the body. When the body is considered deformable, the mesh must be either deformed keeping its topology intact or regenerated changing its topology to conform to the deformable body. The latter is rarely necessary for small deformations. Assuming small local deformations, Piro and Maki (2013) keeps the mesh rigid but still account for the local deformation in the velocity boundary condition. This approach simplifies the mesh update process considerably at the cost of reducing the accuracy of the body boundary condition since the computational mesh is no longer fitted to the deformable body. This approximation may be proved sufficient for many practical applications with small local deformation and the benefit not having to deform the mesh may be compelling. In the present work, however, it is emphasized that the mesh is kept fitted to the deformable body and the body boundary condition is sought to be satisfied accurately.

Solution algorithm for the flow solver

The solution algorithm for the flow solver is presented here to ease the forthcoming discussion on the fluid structure interaction algorithm. The need for the information on the updated location of the fluid-structure interface is emphasized here as this information is essential for the mesh update procedure and for determining the mesh velocity of the ALE frame. Hence, given a new location of the fluid-structure interface (hereafter denoted as \mathbf{x}_{fsi}) the solution algorithm for the fluid-flow solver proceeds as follows:

1. Update mesh according to \mathbf{x}_{fsi} and evaluate mesh velocity, \mathbf{u}_m
2. Update the effective convective velocity $\mathbf{u}_c = \mathbf{u} - \mathbf{u}_m$
3. Solve Eq. (2) for the volume phase fraction α
4. Update \mathbf{u} and p using the PISO algorithm
5. Repeat from 2 until convergence

The mesh update is done explicitly given \mathbf{x}_{fsi} . At time index $n+1$, even though \mathbf{x}_{fsi}^{n+1} is updated through e.g. the iterations of a partitioned FSI (fluid-structure interaction) scheme, the mesh velocity \mathbf{u}_m are evaluated only from \mathbf{x}_{fsi}^n (i.e. the interface location from the previous time index) and the latest provided \mathbf{x}_{fsi}^{n+1} . This constraint exists due to the formulation of the flow solver which assumes that the fluid domain is already known. Hence, any motion of the domain and its boundary must be imposed before carrying out the numerical evaluation of the flow field.

STRUCTURAL SOLVER

There is a need for an accurate model to predict the motions and structural responses of a large flexible vessel in waves. The vessel may experience large rigid body motions (e.g. heave, roll and pitch) while the elastic deformations may be assumed to be small. Indeed, the assumption of small elastic deformations has been the principle argument for neglecting any inertia coupling

between the rigid body motion and the small elastic deformations. Seng et al. (2012) accounts for the flexibility of the hull girder using the classical non-uniform Timoshenko beam. The governing equations for the Timoshenko beam are solved using the classical modal superposition method (Jensen, 2001). The formulation uses dry modes for the modal superposition and the orthogonality between the modes creates a system of the linear differential equations in the modal coordinate. These equations are coupled only through the external forcing terms which contain the contribution from the added mass of water. While this added mass can be approximated by an explicit model, the numerical solution of the free surface CFD can account for it implicitly. The fluid action of the structures is evaluated directly from surface integration of the total fluid pressure with the viscous shear stress included if deemed necessary. The downside is that the implicit evaluation of the added mass comes at a cost of larger computational effort since the implicit evaluation often requires several iterations until the results have converged to a tight tolerance. Nevertheless, the implicit evaluation of the fluid actions on the structural part is one of the key attractors for using the free surface CFD approach.

Different formulations exist to describe the dynamic of a deformable body. These formulations describe the same dynamic system but are different in terms of different frame of references: inertial frame, corotational frame, and floating frame. Wasfy and Noor (2003) provide an excellent review of these formulations. In the inertial frame formulation, the global inertial frame of reference is applied to describe any motions. In the co-rotational frame formulation, a local coordinate system is attached to every element of the finite element model and the motions are described in relation to these local frames. In the floating frame formulation, a local coordinate system is attached to the body and any local deformation on the body is described in relation to this frame. The floating frame formulation is a popular choice (see Shabana, 2010) because it is the most practical to apply in connection with a linear modal superposition technique to reduce the number of equations from several thousands to only a few corresponding to the number of the selected mode shapes. It is also possible to use mode shapes and natural frequencies from experiments directly in the formulation hence avoiding a numerical modelling of the mode shapes. The floating frame formulation has been chosen in this work and the governing equations shown below aim at providing a clear view of the various terms.

Consider any point on the deformable body, the position vector \mathbf{r} written with respect to the earth fixed inertial frame is

$$\mathbf{r} = \mathbf{R} + \mathbf{A}\bar{\mathbf{u}} \quad (5)$$

The notation is adopted from Shabana (2010) where an overline is used to indicate the relation to the local frame. The origin of the local coordinate system is located at \mathbf{R} and the corresponding transformation matrix is \mathbf{A} . In the undeformed state, the position vector of a point written in this local frame is denoted by $\bar{\mathbf{u}}_0$. The local deformation associated with this point is denoted by $\bar{\mathbf{u}}_f$ and approximated by means of the Rayleigh-Ritz method as $\bar{\mathbf{u}}_f \cong \Phi \mathbf{q}_f$, where $\mathbf{q}_f \cong \mathbf{q}_f(t)$ is the vector containing the time-dependent generalized coordinates; Φ is the mode shapes matrix containing k number of deformation modes $[\Phi_1, \Phi_2, \dots, \Phi_k]$. The position vector written in the local frame is written as

$$\bar{\mathbf{u}} = \bar{\mathbf{u}}_0 + \Phi \mathbf{q}_f \quad (6)$$

The time derivative (denoted by an overdot) of \mathbf{r} provides the absolute velocity and the kinetic energy T can be written

$$T = \frac{1}{2} \int_V \dot{\mathbf{r}}^T \dot{\mathbf{r}} \rho dV = \frac{1}{2} \begin{bmatrix} \dot{\mathbf{R}} \\ \bar{\boldsymbol{\omega}} \\ \dot{\mathbf{q}}_f \end{bmatrix}^T \mathbf{M} \begin{bmatrix} \dot{\mathbf{R}} \\ \bar{\boldsymbol{\omega}} \\ \dot{\mathbf{q}}_f \end{bmatrix} \quad (7)$$

where $\bar{\boldsymbol{\omega}}$ is the angular velocity of the orientation angles of the local frame written with respect to the local frame. The dot denotes the time derivative; ρ is the material density and the integral is performed over the volume V . When using the

distributed mass formulation, the matrix \mathbf{M} will become a consistent mass matrix and the integral will depend on the type of the finite elements applied to discretize the body.

$$\mathbf{M} = \int_V \begin{bmatrix} \mathbf{I} \\ -A\tilde{\mathbf{u}} \\ A\Phi \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ -A\tilde{\mathbf{u}} \\ A\Phi \end{bmatrix}^T \rho dV = \int_V \begin{bmatrix} \mathbf{I} & -A\tilde{\mathbf{u}} & A\Phi \\ \dots & \tilde{\mathbf{u}}^T \tilde{\mathbf{u}} & \tilde{\mathbf{u}}^T \Phi \\ sym. & \dots & \Phi^T \Phi \end{bmatrix} \rho dV \quad (8)$$

The system mass matrix is symmetric. Hence, only the upper parts are shown here and the “sym” denotes that the lower part can be obtained by the symmetry. The symbol \mathbf{I} denotes the (3×3) identity matrix. The tilde found in the expression denotes the skew-symmetric matrix operator which is defined as

$$\tilde{\mathbf{u}} = \bar{\mathbf{u}} \times = \begin{bmatrix} 0 & -\bar{u}_3 & \bar{u}_2 \\ \bar{u}_3 & 0 & -\bar{u}_1 \\ -\bar{u}_2 & \bar{u}_1 & 0 \end{bmatrix} \quad (9)$$

The system mass matrix can be simplified considerably if the location of the local coordinate system is selected such that its origin coincides with the center of gravity of the body in the undeformed state. Hence,

$$\int_V \bar{\mathbf{u}}_0 \rho dV = 0 \Rightarrow \int_V \tilde{\mathbf{u}}_0 \rho dV = 0 \quad (10)$$

It shall be emphasized that this equality does not require the origin of the local frame to remain attached at the center of gravity when the body has been deformed. Further simplification is obtained through an appropriate use of the modal superposition principle. The selected mode shapes shall be dry modes corresponding to free vibration in vacuum. The mode shapes are derived under a free vibration without supports i.e. without any surface traction acting on the surface of the body. Under this condition, Sherif et al. (2012) show that the system mass matrix reduces to

$$\mathbf{M} = \int_V \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{u}}^T \tilde{\mathbf{u}} & \tilde{\mathbf{u}}^T \Phi \\ \mathbf{0} & \Phi^T \tilde{\mathbf{u}}^T & \Phi^T \Phi \end{bmatrix} \rho dV \quad (11)$$

Due to the orthogonality of the selected mode shapes, the components $\int_V \Phi^T \Phi \rho dV$ will reduce to a diagonal matrix. When the mode shapes are mass normalized, this diagonal matrix will reduce to the identity matrix. The final equation of motions are derived from the Lagrange's equations

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\mathbf{q}}^T} \right) - \left(\frac{\partial T}{\partial \mathbf{q}^T} \right) + \left(\frac{\partial U}{\partial \mathbf{q}^T} \right) - \mathbf{G} = \mathbf{0} \quad (12)$$

where \mathbf{G} is the generalized external forces and $\mathbf{q} = [\mathbf{R}^T \quad \bar{\boldsymbol{\theta}}^T \quad \mathbf{q}_f^T]^T$ is the generalized coordinates containing the location and orientation of the body frame of reference and the modal coordinates which describe the local deformation on the body. The elastic energy U is written as $\frac{1}{2} \mathbf{q}^T \mathbf{K}_f \mathbf{q}$, where \mathbf{K}_f is the structural stiffness matrix. The final equations of motion can be written in the standard notation as

$$[\mathbf{M}] \ddot{\mathbf{q}} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} \end{bmatrix} \dot{\mathbf{q}} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K} \end{bmatrix} \mathbf{q} = \mathbf{G} + \mathbf{Q} \quad (13)$$

where the system mass matrix is given in Eq. (11) and the structural damping is included in terms of modal damping. The orthogonality of the mode shapes produces diagonal form of the structural stiffness and damping matrices. There is no artificial damping added to the rotational rigid degree of freedom. The damping for the translational or rotational rigid body degree of freedom appears only in terms of the implicit fluid action on the structure. The symbol \mathbf{Q} is the quadratic velocity terms containing the contribution from the centrifugal and the Coriolis effects.

$$\mathbf{Q} = \int_V \begin{bmatrix} -A(\ddot{\bar{\omega}})^2 \bar{\mathbf{u}} - 2A\ddot{\bar{\omega}}\dot{\bar{\mathbf{u}}}_f \\ \ddot{\bar{\omega}}(\ddot{\bar{\mathbf{u}}})^2 \bar{\omega} + 2\ddot{\bar{\mathbf{u}}}\dot{\bar{\omega}} \\ -\Phi^T \left\{ (\ddot{\bar{\omega}})^2 \bar{\mathbf{u}} + 2\ddot{\bar{\omega}}\dot{\bar{\mathbf{u}}}_f \right\} \end{bmatrix} \rho dV = \int_V \begin{bmatrix} 0 \\ \ddot{\bar{\omega}}(\ddot{\bar{\mathbf{u}}})^2 \bar{\omega} + 2\ddot{\bar{\mathbf{u}}}\dot{\bar{\omega}} \\ -\Phi^T \left\{ (\ddot{\bar{\omega}})^2 \bar{\mathbf{u}} + 2\ddot{\bar{\omega}}\dot{\bar{\mathbf{u}}}_f \right\} \end{bmatrix} \rho dV \quad (14)$$

The first row in the above expression vanishes due to the use of dry mode shapes and the local frame attached to the center of gravity at the undeformed state. The generalized external forces \mathbf{G} can be derived using the variational principle of virtual work which yields

$$\mathbf{G} = \int_V \begin{bmatrix} \mathbf{f} \\ \ddot{\bar{\mathbf{u}}} (A^T \mathbf{f}) \\ \Phi^T (A^T \mathbf{f}) \end{bmatrix} dV \quad (15)$$

where \mathbf{f} is the external force distribution acting on the body which contains both the volume force due to the gravitational acceleration and the fluid action on the body. The formulation requires \mathbf{f} to be expressed in the global inertial frame of reference which is very convenient since the surface integration of the pressure and viscous shear stress obtained from e.g. the free surface CFD method are usually readily expressed with respect to the global inertial frame of reference.

From Eqs. (10)–(15), it can be seen that the equations for \mathbf{R} , i.e. the translation of the origin of the local floating frame, are uncoupled from the rest of the system of equations. However, the system mass matrix is time variant and contains an inertia coupling between the structural deformation and the degree of freedoms for the rotational motion of the local floating frame. This coupling is nonlinear and causes additional complexity in the design of the numerical solution algorithm. Therefore, these inertia coupling are commonly neglected and at the same time making the system mass matrix time invariant by neglecting the time dependent mass moment of inertia. The approximate system mass matrix is

$$\mathbf{M} \cong \int_V \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddot{\bar{\mathbf{u}}}_0^T \ddot{\bar{\mathbf{u}}}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Phi^T \Phi \end{bmatrix} \rho dV \quad (16)$$

This approximation is equivalent to the approximation where the small local deformation is superimposed on the gross rigid body motions with the exception that the influence of the local vibrations on the rigid body rotations is retained within the quadratic velocity terms \mathbf{Q} . In the present paper, however, a numerical solution algorithm has been designed to solve the equations of motion in the form as presented in Eq. (11) i.e. all components of the system mass matrix are accounted for.

Evaluation of the orientation of the local floating frame

The orientation matrix \mathbf{A} is required in Eq. (15) to transform the force vector \mathbf{f} into the local floating frame of reference. In this work the rotational coordinates are described by Euler parameters, $[e_0 \ e_1 \ e_2 \ e_3]^T$. The use of Euler parameters has an advantage over the Euler angles since problems with singularity does not exist. However, the four Euler parameters must

always obey the unity constrain; i.e. $\mathbf{e}^T \mathbf{e} = 1$. The time derivative of this constrain is $\mathbf{e}^T \dot{\mathbf{e}} = 0$. The relation between these Euler parameters and $\bar{\boldsymbol{\omega}}$ (the angular velocity expressed in the local frame) is cf. Shabana (2010),

$$\dot{\mathbf{e}} = \frac{1}{2} \bar{\mathbf{E}}^T \bar{\boldsymbol{\omega}} = \frac{1}{2} \mathbf{H}(\bar{\boldsymbol{\omega}}) \mathbf{e} \quad (17)$$

where the matrix $\bar{\mathbf{E}}$ and the operator \mathbf{H} are defined as

$$\bar{\mathbf{E}} = \begin{bmatrix} -e_1 & e_0 & e_3 & -e_2 \\ -e_2 & -e_3 & e_0 & e_1 \\ -e_3 & e_2 & -e_1 & e_0 \end{bmatrix} \quad ; \quad \mathbf{H}(\bar{\boldsymbol{\omega}}) = \begin{bmatrix} 0 & -\bar{\boldsymbol{\omega}}^T \\ \bar{\boldsymbol{\omega}} & -\dot{\bar{\boldsymbol{\omega}}} \end{bmatrix} = \begin{bmatrix} 0 & -\bar{\omega}_1 & -\bar{\omega}_2 & -\bar{\omega}_3 \\ \bar{\omega}_1 & 0 & \bar{\omega}_3 & -\bar{\omega}_2 \\ \bar{\omega}_2 & -\bar{\omega}_3 & 0 & \bar{\omega}_1 \\ \bar{\omega}_3 & \bar{\omega}_2 & -\bar{\omega}_1 & 0 \end{bmatrix}$$

Eq. (17) must be carefully integrated to avoid the violation of the unity constraint. In the present work, a second order midpoint integration scheme is designed to obey the unity constraint exactly. Using the definition $\bar{\boldsymbol{\omega}} = \dot{\bar{\boldsymbol{\theta}}}$, Eq. (17) can be integrated by part from time n to $n+1$. The result is

$$\Delta \mathbf{e} = \Delta \left[\frac{1}{2} \bar{\mathbf{E}}^T \bar{\boldsymbol{\theta}} \right] - \frac{1}{2} \int_n^{n+1} \bar{\mathbf{E}}^T \bar{\boldsymbol{\theta}} dt \quad (18)$$

The delta-operator Δ is defined here as the increment value between time n and $n+1$. For instant, $\Delta \mathbf{e} = \mathbf{e}_{n+1} - \mathbf{e}_n$. The result presented in Eq. (18) is exact. However, a numerical evaluation requires an approximation to the last term. The choice here is that the variation of $\bar{\boldsymbol{\theta}}$ over the time interval is approximated as $(\bar{\boldsymbol{\theta}}_n + \bar{\boldsymbol{\theta}}_{n+1})/2$. Applying this approximation, a simple arithmetic manipulation yields an update scheme written as

$$\mathbf{e}_{n+1} = \frac{(16 - \Delta \bar{\boldsymbol{\theta}}^T \Delta \bar{\boldsymbol{\theta}}) \mathbf{I}_4 + 8 \mathbf{H}(\Delta \bar{\boldsymbol{\theta}})}{16 + \Delta \bar{\boldsymbol{\theta}}^T \Delta \bar{\boldsymbol{\theta}}} \mathbf{e}_n \quad (19)$$

This is an explicit scheme where the four Euler parameter can be updated given the increment $\Delta \bar{\boldsymbol{\theta}}$ which can be obtained directly from a numerical integration of $\bar{\boldsymbol{\omega}} = \dot{\bar{\boldsymbol{\theta}}}$. The symbol \mathbf{I}_4 is the (4 x 4) identity matrix and the \mathbf{H} operator defined previously has been applied to simplify the expression. It can be easily verified by inserting Eq. (19) in the constraint equation to show that the following recursive expression is valid for any increment of $\Delta \bar{\boldsymbol{\theta}}$

$$\mathbf{e}_{n+1}^T \mathbf{e}_{n+1} = \mathbf{e}_n^T \mathbf{e}_n \quad (20)$$

Therefore, the unity constraint is satisfied exactly given that the initial value for the four Euler parameters is valid and obeys the unity constraint exactly. Once the four Euler parameters have been evaluated, the orientation matrix \mathbf{A} can be evaluated as shown in Shabana (2010).

A numerical solution of the coupled equations of motion

A numerical solution algorithm is sought for the following system of nonlinear ordinary differential equations

$$\int_V \rho dV \ddot{\mathbf{R}} = \int_V \mathbf{f} dV \quad (21)$$

$$\int_V \begin{bmatrix} \tilde{\mathbf{u}}^T \tilde{\mathbf{u}} & \tilde{\mathbf{u}}_f^T \Phi \\ \Phi^T \tilde{\mathbf{u}}_f & \Phi^T \Phi \end{bmatrix} \rho dV \begin{bmatrix} \dot{\tilde{\omega}} \\ \dot{\tilde{\mathbf{q}}}_f \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \tilde{\omega} \\ \tilde{\mathbf{q}}_f \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K} \end{bmatrix} \begin{bmatrix} \tilde{\theta} \\ \tilde{\mathbf{q}}_f \end{bmatrix} = \int_V \begin{bmatrix} \tilde{\mathbf{u}}^T (A^T \mathbf{f}) \\ \Phi^T (A^T \mathbf{f}) \end{bmatrix} dV + \int_V \begin{bmatrix} \tilde{\omega} (\tilde{\mathbf{u}})^2 \tilde{\omega} + 2\tilde{\mathbf{u}}_f^T \tilde{\omega} \\ -\Phi^T \left\{ (\tilde{\omega})^2 \tilde{\mathbf{u}} + 2\tilde{\omega} \tilde{\mathbf{u}}_f \right\} \end{bmatrix} \rho dV \quad (22)$$

Eq. (21) shall be treated as a nonlinear systems since the added mass of water will be evaluated implicitly within the external force distribution \mathbf{f} . With an iterative algorithm, it will be necessary to introduce an under-relaxation scheme into the algorithm to prevent numerical instability which can occur when there is a strong added-mass effect; see e.g. Causin et al. (2005). A predictor-corrector scheme e.g. a second order Adam Bashforth Moulton method or the classical 4th-order Runge-Kutta scheme can be applied directly to Eq. (21). However, it is important to mention that the evaluation of \mathbf{f} can be an extremely resource-demanding task; especially with a finite volume free surface CFD method. Therefore, the intermediate solutions required by the Runge-Kutta scheme may need to be evaluated using approximated value for \mathbf{f} e.g. through interpolation which will unavoidably reduce the order of the Runge-Kutta scheme.

Eq. (22) is rearranged to exploit the diagonal structures of damping and stiffness matrices as well as the terms $\int_V \Phi^T \Phi \rho dV$ which will become an identity matrix when using mass normalized mode shapes. With simple manipulations Eq. (22) can be rewritten as

$$\int_V (\tilde{\mathbf{u}}^T \tilde{\mathbf{u}} - \tilde{\mathbf{u}}_f^T \Phi \Phi^T \tilde{\mathbf{u}}_f) \dot{\tilde{\omega}} \rho dV = \int_V \tilde{\mathbf{u}}^T (A^T \mathbf{f}) dV + \int_V \left[\tilde{\omega} (\tilde{\mathbf{u}})^2 \tilde{\omega} + 2\tilde{\mathbf{u}}_f^T \tilde{\omega} \right] \rho dV - \int_V \tilde{\mathbf{u}}_f^T \Phi \hat{\tilde{\mathbf{q}}}_f \rho dV \quad (23)$$

$$\int_V \Phi^T \Phi \rho dV \ddot{\tilde{\mathbf{q}}}_f = \hat{\tilde{\mathbf{q}}}_f - \int_V \Phi^T \tilde{\mathbf{u}}_f^T \dot{\tilde{\omega}} \rho dV \quad (24)$$

where $\hat{\tilde{\mathbf{q}}}_f$ in the above expressions is defined as

$$\hat{\tilde{\mathbf{q}}}_f = \int_V \Phi^T (A^T \mathbf{f}) dV + \int_V -\Phi^T \left\{ (\tilde{\omega})^2 \tilde{\mathbf{u}} + 2\tilde{\omega} \tilde{\mathbf{u}}_f \right\} \rho dV - \mathbf{C} \dot{\tilde{\mathbf{q}}}_f - \mathbf{K} \tilde{\mathbf{q}}_f$$

A numerical solution for Eq. (22) is challenging to formulate since the system mass matrix may become badly conditioned due to the fact that the total mass moment of inertia is usually several order of magnitudes larger than the identity matrix of the terms $\int_V \Phi^T \Phi \rho dV$. A scaling can be applied to the equations to improve the conditioning number. In the present work, however, the approach is to solve Eq. (22) in two sequential steps using the formulation shown in Eq. (23) and (24). In the first step, Eq. (23) is solved by imposing the latest known $\tilde{\mathbf{q}}_f$ and $\dot{\tilde{\mathbf{q}}}_f$. The solution of Eq. (23) provides an estimate of the angular acceleration $\dot{\tilde{\omega}}$ needed for solving Eq. (24). Eqs. (23) and (24) are solved iteratively in this sequence using e.g. the second order Adam Bashforth Moulton method.

STRONGLY COUPLED FSI SCHEME

The FSI algorithm applied in the present work can be classified as a strongly coupled partitioned scheme which operates in a predictor-corrector manner. Within the corrector step, an inner iteration with under-relaxation is introduced to satisfy the coupling condition at the interface. The partitioned approach lets the flow solver and the structural solver to operate alternately until a dynamic equilibrium is achieved. The alternating sequence is very flexible in a sense that it provides the possibility to apply a wide range combination of flow and structural solvers. For instant, the present structural solver can be replaced by a full FEM analysis or the classical non-uniform Timoshenko beam model.

Both the flow and the structural solvers are, however, mutually reliant on each other. The solution process for the flow solver requires information on the location of the interface (hereafter denoted as \mathbf{x}_{fsi}) provided by the structural solver. Vice versa, the solution algorithm of the structural solver needs the fluid force distribution on the interface (hereafter denoted as \mathbf{f}_{fsi}).

Consider the solution at time index $n+1$. The fluid force distribution \mathbf{f}_{fsi}^n and the location of the interface \mathbf{x}_{fsi}^n at time index n are known and assumed to satisfy the dynamic equilibrium. The FSI solution algorithm starts by estimating \mathbf{x}_{fsi}^{n+1} , i.e. the location of the interface at time index $n+1$. At this stage the structural solver provides an explicit solution of the displacement which involves executing the structural solver based on the known force distribution from the previous time step. The algorithm implemented in the present work performs a second order extrapolation on the force distribution to obtain the first estimate of \mathbf{f}_{fsi}^{n+1} . The choice to include this extrapolation is motivated by the expectation that the numerical solution of \mathbf{f}_{fsi}^{n+1} from the flow solver can be several orders of magnitude slower than the solution process to obtain \mathbf{x}_{fsi}^{n+1} . A good first guess for \mathbf{f}_{fsi}^{n+1} may help to reach convergence faster.

Once, the very first estimate of \mathbf{x}_{fsi}^{n+1} is known the algorithm proceeds on to obtain a new estimate of \mathbf{f}_{fsi}^{n+1} . This new estimate of \mathbf{f}_{fsi}^{n+1} is now no longer a simple extrapolation but it is evaluated from executing the flow solver based on the information on \mathbf{x}_{fsi}^{n+1} . The process iterates until the changed in \mathbf{x}_{fsi}^{n+1} is smaller than a very tight tolerance. To obtain a good numerical stability it is important to introduce a fixed or a dynamic under-relaxation within this iteration. There is a strong need to keep the number of iterations as small as possible; especially due to the use of the free surface CFD method for the fluid flow. Hence, the fixed under-relaxation is considered inferior to the dynamic one. Here, the classical Aitken's algorithm (Irons and Tuck, 1969) has been applied to accelerate the convergence. The relaxation of the Aitken's algorithm is applied to \mathbf{f}_{fsi}^{n+1} after 3 estimates of its value have been obtained. The extrapolation performed within the predictor step counts as one. The first iteration of the corrector step counts as two. Hence, the Aitken's algorithm can be applied already on the second iteration of the corrector steps and beyond. The scheme is

$$\left(\mathbf{f}_{fsi}^{n+1}\right)^{k+1} = \left(\mathbf{f}_{fsi}^{n+1}\right)^k + w^{k+1} \delta^{k+1} \quad (25)$$

where w is the adaptive relaxation coefficient and the superscript " k " is the index number of the iteration. Initially, the under-relaxation coefficient is set to 1 and the subsequence update is evaluated as

$$w^{k+1} = \min \left[2, \max \left(0, -\frac{(\delta^{k+1} - \delta^k)^T \delta^k}{(\delta^{k+1} - \delta^k)^T (\delta^{k+1} - \delta^k)} w^k \right) \right] \quad (26)$$

where the δ -operator is defined as

$$\delta^{k+1} = \left(\mathbf{f}_{fsi}^{n+1}\right)^{k+1} - \left(\mathbf{f}_{fsi}^{n+1}\right)^k$$

The $\min()$ and $\max()$ functions limit the relaxation coefficient w to interval $[0, 2]$. The value of w is allowed to exceed 1 up to 2 since an over-relaxation is possible. The present FSI scheme can be summarized as follows:

The predictor:

1. iteration $k = 0$: extrapolate for $\left(\mathbf{f}_{fsi}^{n+1}\right)^0$
2. use $\left(\mathbf{f}_{fsi}^{n+1}\right)^k$ in the structural solver to obtain $\left(\mathbf{x}_{fsi}^{n+1}\right)^k$
3. use $\left(\mathbf{x}_{fsi}^{n+1}\right)^k$ in the flow solver to obtain a new estimate $\left(\mathbf{f}_{fsi}^{n+1}\right)^{k+1}$
4. check for convergence: done if $\left\| \left(\mathbf{f}_{fsi}^{n+1}\right)^1 - \left(\mathbf{f}_{fsi}^{n+1}\right)^0 \right\|_2 < tolerance$

The corrector:

5. iteration $k = 1$: repeat step 2 & 3 to obtain $(\mathbf{x}_{fsi}^{n+1})^1$ and $(\mathbf{f}_{fsi}^{n+1})^2$
6. check for convergence: done if $\|(\mathbf{x}_{fsi}^{n+1})^1 - (\mathbf{x}_{fsi}^{n+1})^0\|_2 < tolerance$
7. iteration $k = 2$: relax $(\mathbf{f}_{fsi}^{n+1})^2$ i.e. Eqs. (25) ~ (26) and repeat step 2 & 3 to obtain $(\mathbf{x}_{fsi}^{n+1})^2$ and $(\mathbf{f}_{fsi}^{n+1})^3$
8. check for convergence: done if $\|(\mathbf{x}_{fsi}^{n+1})^k - (\mathbf{x}_{fsi}^{n+1})^{k-1}\|_2 < tolerance$, otherwise increases k and repeats from 7

This predictor-corrector scheme is applied at every time step. The performance of the algorithm and the number of the iteration required to reach the converged solution may depend strongly on the effect of the added-mass, the size of the chosen time step and the specified tolerance. In the numerical example presented in this paper, the iteration converges at $k = 3 \sim 5$. The algorithm converges rarely at $k = 0$, therefore the convergence check based on the force distribution at step 4 can be skipped.

FSI INTERFACE

The FSI interface acts as a platform for transferring information between both solvers. On the fluid side the fluid force distribution is calculated and transferred to the structural side. The structural solver receives the force distribution and calculates the displacement of the interface which is then transferred back to the fluid side. This transferring operation is complicated by the often very different resolutions of the discretization applied in both solvers. Typically, the finite volume discretization in the flow solver requires a much finer resolution than the finite element discretization applied in the structural solver. Furthermore, the level of modelling detail may not be the same. Consider a large vessel, for example, the structural solver may include not only the hull but also superstructure in the model. The flow solver, on the other hand, does not need to model the superstructure since the fluid action on the vessel is mainly related to the wetted surface of the hull. This may seem easy to overcome since one can simply select the wetted surface of the hull to be the FSI interface. However, the mesh update procedure in the flow solver needs not only the displacement of the wetted surface but also the displacement of the deck floor which may have been simplified by a flat cover for the wetted hull and therefore does not match with the FEM model exactly. In some cases, not even the wetted surface is matching between the two solvers. This missing information must be determined e.g. through interpolations or extrapolations in three dimensional spaces. The interpolation process is further complicated by the irregularity of the unstructured grid. The transferred information shall satisfy the coupling conditions which are

$$(\mathbf{x}_{fsi})_S = (\mathbf{x}_{fsi})_F \quad (27)$$

$$(\dot{\mathbf{x}}_{fsi})_S = (\dot{\mathbf{x}}_{fsi})_F \quad (28)$$

$$(\boldsymbol{\sigma}_{fsi} \cdot \mathbf{n})_S = (\boldsymbol{\sigma}_{fsi} \cdot \mathbf{n})_F \quad (29)$$

where the subscripts "S" and "F" denote the structural and the fluid mesh respectively. The surface unit normal vector of the FSI interface is denoted by \mathbf{n} and the surface stress tensor is $\boldsymbol{\sigma}$. The first two conditions, Eqs. (27) ~ (28), express the kinematic continuity across the FSI interface, i.e. displacement and velocity of the FSI interface shall match exactly. The last condition, Eq. (29), expresses the dynamic continuity which requires that forces acting on the FSI interface shall be exactly the same on the two meshes. These conditions are satisfied when displacement of the FSI interface and the force distribution are transferred using a fully energy conserving method. The transferring operation can be done through e.g. an energy conserving grid-to-grid mapping; however, it is worth noticing that the force distribution projected into the modal spaces does not require a mapping operation.

Transferring the nodal displacement and velocity

The structural solver provides the nodal displacement and velocity of the FSI interface on the structural mesh. The objective is to determine the equivalent displacement and velocity on the fluid mesh. In the global inertial frame, any node on the structural mesh is determined by Eq. (5). The same formulation can be applied to the nodes of the fluid mesh. Therefore, it is sufficient to transfer the mode shapes to the fluid mesh. This is a one-time transfer operation done at the initialization phase of the simulation. The mode shapes are transferred from the structural mesh to the fluid mesh using a grid-to-grid mapping method.

The grid-to-grid mapping will depend strongly on the type of the finite element used in the structural solver. For example, a mapping from a beam element is very different from a mapping from a shell element. The mapping procedure described here assumes that the FSI interfaces in both meshes are given as triangulated surfaces. Hereafter the triangulated surface of the FSI interface on the structural mesh will be called “the source” and on the fluid mesh “the target”. Both the source and the target surfaces are generally considered non-matching and they are assumed to be overlapping.

Ideally, the total coverage of the source and the target surfaces is exactly the same. This is possible if the FSI interface is a planar surface. For a general non-planar surface the coverage is not necessarily the same due to the different resolution of discretization. Therefore, a projection of the target points onto the source surface is performed. Then, the local triangle on the source surface is identified for each of the projected points for the purpose of interpolation. In the present work this interpolation is done linearly over each local triangle. The error induced in this transferring process will affect the implicit evaluation of the added mass of water. In a comparison to other approaches where the added mass is evaluated at infinitely high frequency, it is reasonable to presume that the first order accuracy obtained from the present linear interpolation is sufficient. Nevertheless, a higher-order interpolation can be very well applied here and one can easily justify the need for a higher-order scheme; e.g. the order of the coupling schemes cannot exceed the order of the interpolation scheme. Hence, the linear interpolation selected in the current work will limit the accuracy of coupled solutions to first order.

The point projection method described here is not versatile enough to be used in all simulation cases since it requires that the source and target surfaces are fully overlapping. In the case where a partial overlapping occurs the projected points may lie outside the coverage area of the source surface causing a failure when identifying the triangle for the interpolation. A nearest triangle cannot be used here since an extrapolation (which is much less reliable) must be performed. As an alternative to this point projection method, a general point cloud interpolation in three dimensional space based on radial basis functions (RBF, Bos et al., 2013) could be selected. The interpolation based on RBF is well known for being versatile and have been used for representing a three dimensional body where the provided mesh is incomplete; see e.g. Carr et al. (2001).

Transferring the force distribution

The equations of motion need the force distribution \mathbf{f} from the flow solver. Farrell and Maddison (2011) provide a very efficient grid-to-grid transfer based on the local Galerkin projection method. An implementation of the algorithm is provided by the OpenFOAM software package and has been considered in the present work. A grid-to-grid transfer must satisfy the dynamic continuity on the FSI interface presented in Eq. (29) and the algorithm presented by Farrell and Maddison 2011 works well for the case where the source and target surfaces are fully overlapping. In the partial overlapping case, the implemented algorithm loses its energy conserving property. This can be easily confirmed by comparing the total force and total moment evaluated on the fluid mesh and the structural mesh.

Eq. (21) requires the force distribution for the integral $\int_V \mathbf{f} dV$ which is basically the total force on the structure. This total force can be evaluated in the fluid mesh. Similarly, Eq. (22) requires \mathbf{f} for evaluating the forcing terms $\int_V \tilde{\mathbf{u}}^T (\mathbf{A}^T \mathbf{f}) dV$ and $\int_V \Phi^T (\mathbf{A}^T \mathbf{f}) dV$ which are projections of the force distribution onto the modal spaces. This term can be evaluated in the fluid mesh as well. Hence, there is no need for a direct transfer of \mathbf{f} . This indirect transfer of the force distribution provides a very competing alternative to the grid-to-grid transfer. The only drawback seems to be that it is less efficient to evaluate these integrals in the fluid mesh since this mesh can be much finer than the structural mesh. Moreover, the nonlinear dependency between the rotation degree-of-freedom and the flexural modal amplitudes may require these integrals to be evaluated several times within each request for a structural solution. Nevertheless, the performance lost may be negligible in a comparison to the overall performance of the flow solver.

NUMERICAL EXAMPLE

The present hydroelastic code is applied to simulate a flexible barge which was presented in Malenica et al. (2003). The experimental model is made of 12 buoyance pontoons attached to 6 mm steel plates at the deck level. The length, width and height of the pontoon are 1.9 m, 0.6 m, and 0.25 m, respectively. These dimensions hold for all pontoons except the foremost pontoon which is different as sketched in Fig. 1. The clearance between each pontoon is 0.015 m creating an overall length of 2.445 m and a draught of 0.12 m. Measured responses under decay tests are presented in Malenica et al. (2003) in terms of vertical displacement at 6 locations along the barge; see Fig. 1. These decay tests were performed by lifting the foremost pontoon to a prescribe level and then release it spontaneously to let the barge floats and vibrates transiently. In the present numerical simulation, the modal coefficients corresponding to the initial configuration are obtained by applying a least square fitting formulated from Eq. (5),

$$\begin{bmatrix} A^T & \Phi_1 \\ A^T & \Phi_2 \\ \vdots & \vdots \\ A^T & \Phi_f \end{bmatrix} = \begin{bmatrix} R \\ q_f \end{bmatrix}_{ic} = \begin{bmatrix} A^T r_1 - \bar{u}_{01} \\ A^T r_2 - \bar{u}_{02} \\ \vdots \\ A^T r_j - \bar{u}_{0j} \end{bmatrix} \Rightarrow \begin{bmatrix} R \\ q_f \end{bmatrix}_{ic} = \left(\begin{bmatrix} A^T & \Phi_1 \\ A^T & \Phi_2 \\ \vdots & \vdots \\ A^T & \Phi_f \end{bmatrix}^T \begin{bmatrix} A^T & \Phi_1 \\ A^T & \Phi_2 \\ \vdots & \vdots \\ A^T & \Phi_f \end{bmatrix} \right)^{-1} \begin{bmatrix} A^T & \Phi_1 \\ A^T & \Phi_2 \\ \vdots & \vdots \\ A^T & \Phi_f \end{bmatrix}^T \begin{bmatrix} A^T r_1 - \bar{u}_{01} \\ A^T r_2 - \bar{u}_{02} \\ \vdots \\ A^T r_j - \bar{u}_{0j} \end{bmatrix} \quad (30)$$

where the subscript “ic” denotes the estimated initial configuration and the integer “j” denotes the number of the available measurements. The information on the rigid body rotation is provided within the transformation matrix A ; hence Eq. (30) is a non-linear least-square fitting problem. The solution provides not only the modal coefficients q_f but also the position of the local coordinate R and the corresponding transformation matrix A . The estimated initial displacements are shown in Fig. 2.

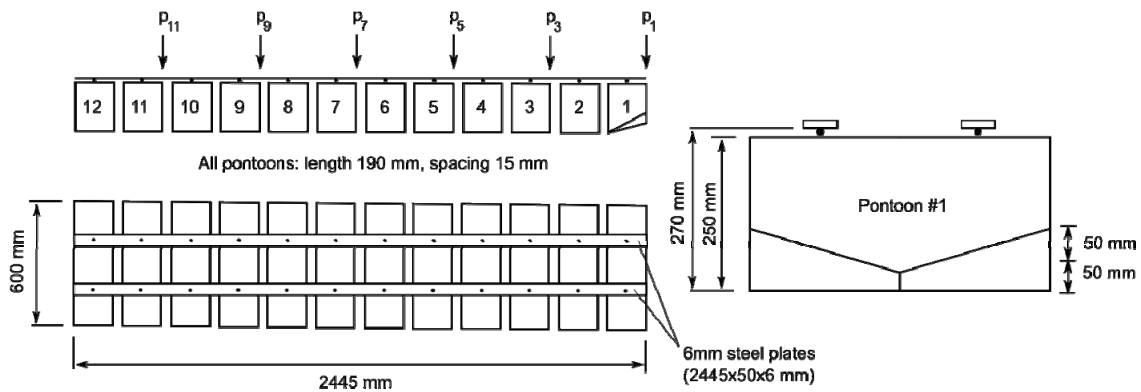


Fig. 1 A sketch of the flexible barge, cf. Malenica et al. (2003) and Remy et al. (2006). Vertical displacement are measured at 6 locations: $p_1(2445 \text{ mm})$, $p_3(2035 \text{ mm})$, $p_5(1625 \text{ mm})$, $p_7(1215 \text{ mm})$, $p_9(805 \text{ mm})$, $p_{11}(395 \text{ mm})$.

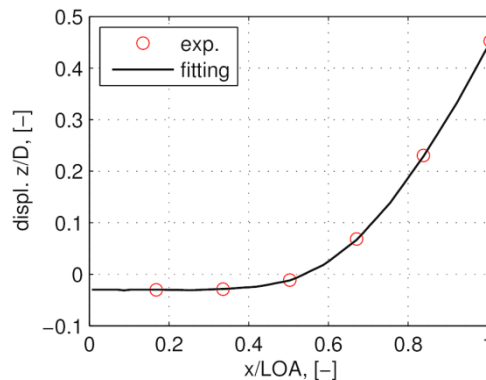


Fig. 2 The least-squared fits of the initial vertical displacement (normalized by draught $D = 0.12 \text{ m}$ and $LOA = 2.445 \text{ m}$).

A FEM model of the flexible barge consisted of beam and shells elements is created in the commercial FEM software ANSYS and a modal analysis of the FEM model is performed with a lumped mass approximation. Fig. 3 shows the 2-node and 3-node modes where it is observed that the pontoons are discretized with shell elements and connected rigidly to the steel plates without any contact between them. The gaps between the pontoons (15 mm) are limiting the allowable local deflection that can be transferred to the CFD mesh. In total, 9 flexural dry modes are selected to describe the local deformation of the barge. Only the vertical vibrational modes are selected. The numerically predicted eigenfrequencies for 2–4-node modes are presented in Table 1. The experimental values for the damping ratio associated with dry modes are not available. In the present simulation the modal damping for the 2-node mode is set to 0.5% of the critical damping ratio. This damping ratio is purely for the internal structural damping which for the current setup is due to the structural damping in the two steel plates at the deck level. The damping associated with the hydrodynamic of the fluid including the additional damping which may appear due to the small gaps between the pontoons is captured implicitly in the free surface CFD flow solver.

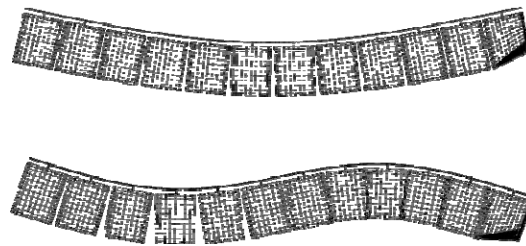


Fig. 3 2-node (top) and 3-node (bottom) dry mode shapes evaluated by the FEM model. There is no contact constraint in the FEM model. Hence, the small gaps (15 mm) between the pontoons are limiting the allowable deflection which can be transferred to the CFD mesh.

Table 1 Numerically predicted dry-mode eigenfrequencies.

Vertical dry modes	Eigenfreq., [Hz]
2-node	0.94
3-node	2.34
4-node	4.02

The CFD mesh is created with unstructured split-hexahedra cells. The hexahedral cells in the far-field region have an edge length of 0.2 m. A cell refinement inside the domain is defined in terms of refinement level relative to this edge length. A refinement level n , by this definition, has an edge length of $0.2m/2^n$, e.g. a refinement level 3 has an edge length of approximately $0.2/2^3 = 0.025$ m. A refinement of level 5 (an edge length of approximately 0.625 mm) is set on the surface of the barge. The cells in the vicinity of the barge are refined based on their distance to the barge. In total, four regions are refined at distances: 0.1 m (level 5), 0.15 m (level 4), 0.5 m (level 3), and 1.5 m (level 2). The free surface cells are refined to level 3 in the horizontal directions and up to level 6 in the vertical direction. With this configuration, the smallest free surface cells have a cell height of 3.12 mm. Lengthwise, there are two layers of cells in the gaps between the pontoons. Due to the symmetry, only half of the barge is modelled and the total number of cells in the domain is 1.26M cells where 90k of these cells are located on the surface of the barge. With the same configuration, two more refined meshes are generated by simply reducing the edge length of the far-field region from 0.2 m to 0.175 m and 0.15 m which yield the two more meshes of sizes 1.72M cells and 3.07M cells, respectively. The results are presented in Figs. 6, 7 and 8.

The overall size of the computational domain is 4.5 times LOA in length, 0.75 times LOA in width and 0.8 times LOA in height. A size view of the CFD mesh is presented in Fig. 4. The gaps between the pontoons are kept in the CFD mesh such that the surfaces which represent the barge in the CFD mesh and in the FEM mesh are fully overlapping. Hence, the point projection method and the linear triangular interpolation are applicable and have been applied for interpolating the selected mode shapes from the FEM surface mesh to the CFD surface mesh.

The spatial scheme selected for the fluid solver is a 2nd-order upwind scheme. However, the 1st-order implicit Euler has been selected for the temporal scheme due to its favorable numerical stability. The size of the time step is adjusted adaptively based on a maximum local Courant number of 0.6. This choice produces a variation of time step size between 0.001 *sec.* ~ 0.0035 *sec.* The simulation runs in parallel on a modern HPC cluster using 32 computed cores from 4 nodes where each computed node consists of two Intel Xeon X5550 (@2.67 GHz) CPUs. Depending on the required number of FSI iterations, the execution time needed to complete each time step vary between 15 *sec.* ~ 40 *sec.* (for mesh 1.72M). The total execution time needed to complete a simulation of 5 *sec.* physical time is approximately 18 hours for mesh 1.72M.

A snapshot of the free surface (at time 0.5 *sec.* after the barge has been released) is presented in Fig. 5 showing the radiation waves generated upon the release of the barge. These waves propagate rapidly toward the boundary of the computational domain. A reflection of these waves will interfere with the flow field around the barge. In the simulation relaxation zones have been attached to the boundaries of the computational domain with the purpose to absorb these radiation waves. The boundaries to the front and to the rear of the barge have a relaxation zone of 1.5 *m* in length attached to them. The boundary to the port side of the barge has a relaxation zone of 0.6 *m*. Further details on the methodology and the performance of the relaxation zone are presented in Jacobsen et al. (2012).

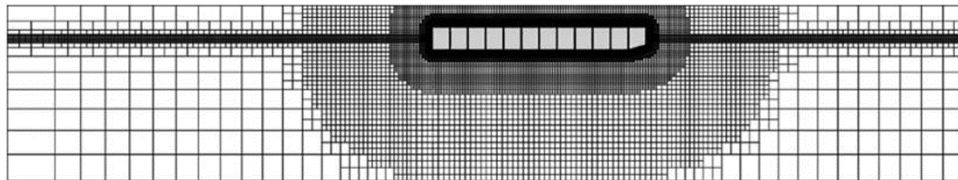


Fig. 4 A side view of the fluid mesh. Three meshes have been generated using this configuration: 1.26M, 1.72M and 3.07M cells.

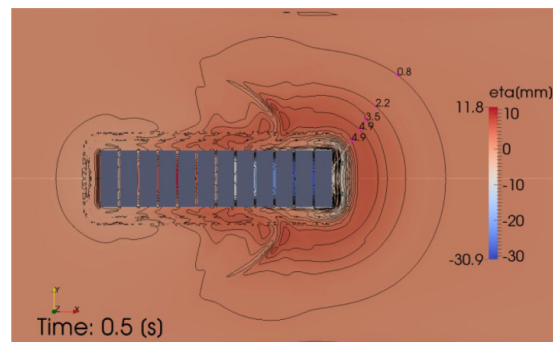


Fig. 5 A snapshot (at time 0.5 *sec.*) of the free surface elevation showing waves radiating away from the barge when released. Due to symmetry the simulation accounts for half of the barge only, and here, the results are post-processed to show both sides of the symmetry plane. The isolines show the free surface elevation in millimeters.

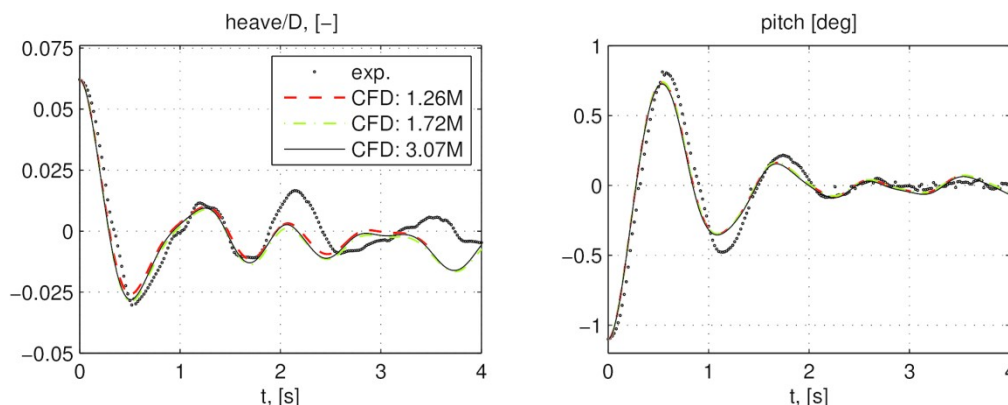


Fig. 6 Simulated heave and pitch compared to the measurements. Heave is normalized by the draught $D=0.12$ *m*.

In Figs. 6 and 7, the time series of the calculated heave, pitch and modal coefficients q_f are compared to their experimental equivalents. The experimental values for heave and pitch are not directly measured, but extracted from the measurements of the vertical displacements at the 6 locations (see Fig. 1) using the least-square fitting method, Eq. (30), applied with the current numerically estimated mode shapes. Hence, the presented experimental values for heave, pitch and q_f are dependent on the supplied numerical mode shapes. On Fig. 6, it can be clearly observed that although there is a similar trend, the predicted heave is slightly smaller than the measurement during the first cycle of the transient vibration. The same is true for the pitch angles. These discrepancies may be related to the fact that the supplied natural frequencies of the 3-node modes and higher (see Fig. 7) may not agree with the experiments. Indeed, a correct evaluation of the natural frequencies will require a better knowledge of the structural details for the pontoons and the two steel plates. In the present simulation the mass distribution is estimated purely from the 0.12 m draught and the stiffness of the beams are adjusted such that the natural frequency of the 2-node mode agrees. The numerically predicted natural frequency for the 3-node mode is found to be too low.

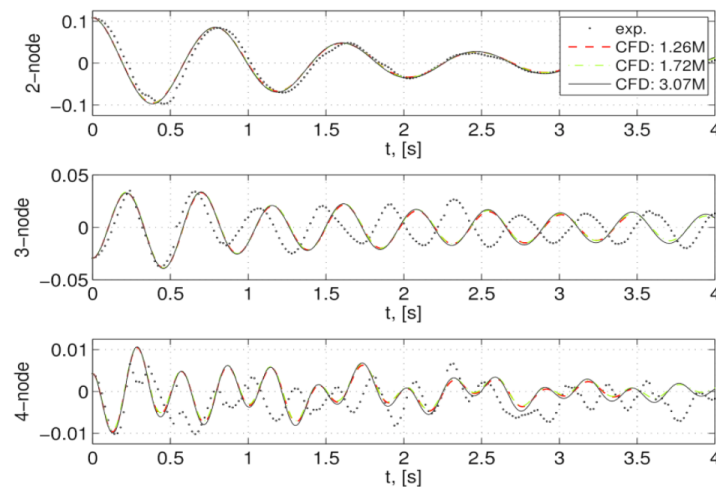


Fig. 7 Numerical results of the modal coefficients compared with the values extracted from the measurements.

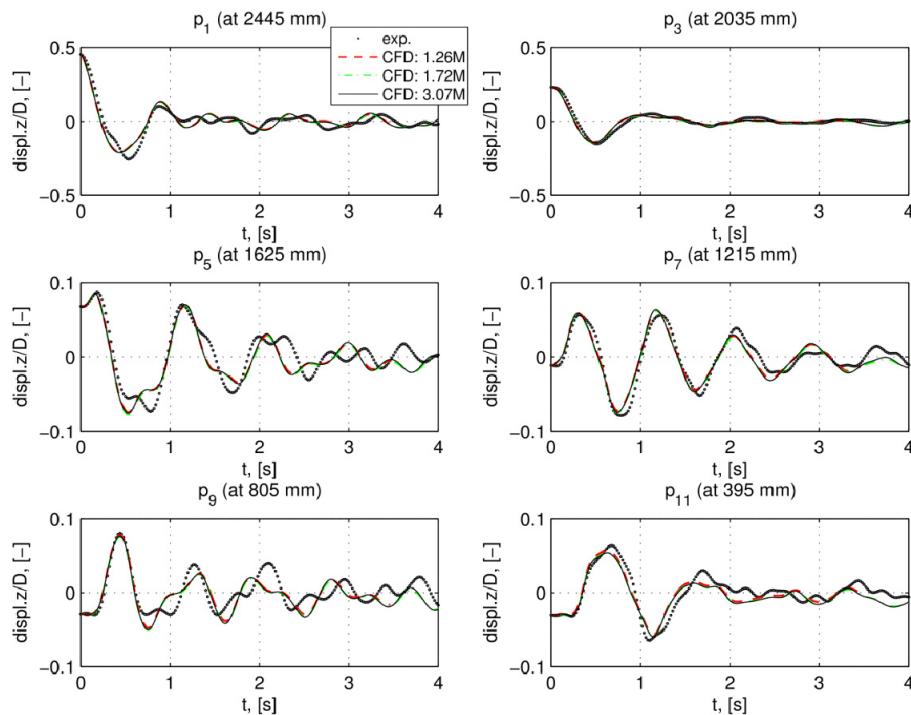


Fig. 8 Vertical displacements measured at 6 locations (see Fig. 1); the vertical displacements are normalized by the draught $D=0.12$ m.

On the other hand, the amplitudes of the modal coefficients agree reasonably well with the measurements. This implies that the imposed structural damping (0.5%) is sufficiently accurate and the fluid damping which are evaluated implicitly within the forcing terms from the flow solver are accurately predicted. However, due to the implicit nature of the numerical procedure it is not practical to quantify the hydrodynamic damping for further analysis.

Fig. 8 shows the vertical displacement measured at 6 positions along the barge at the deck level (as shown in Fig. 1). The locations p_3 , p_7 , and p_{11} are close to the vibrational node of the 3-node mode. Therefore, the contribution from the 3-node mode to the deflection at these locations is limited. It is clear that the agreement between the calculated and the measured displacements is generally good here and much better than other locations (p_1 , p_5 , and p_9) where the influence of the 3-node mode is strong.

CONCLUSIONS

A time-domain hydroelastic code for evaluating the global hydroelastic responses on flexible vessels has been developed by combining a VOF-based free surface flow solver and a flexible-body motion solver in a strongly couple partitioned FSI scheme. The motion solver approximates the local linear elastic deformation of the structure through a modal superposition of the dry mode shapes and preserves the nonlinearity in the rigid body motion and its coupling with the local deformation. A numerical example is given on a simulation of a flexible floating barge undergoing a transient vibration due to a non-zero initial displacement. The numerical results agree satisfactory with the measurements; especially the responses associated with the 2-node mode due to the correct prediction of the natural frequency. These results show that the present hydroelastic code has the potential to accurately predict the global hydroelastic responses of vessels including slamming and springing. There is, however, not yet enough evidence to conclude that this potential has been realized. Further work shall be concentrated on a more systematic validation study in a challenging environment such as e.g. hydroelastic responses in waves where slamming and springing responses are likely to occur.

ACKNOWLEDGEMENTS

The presented work is funded by a join research project between Technical University of Denmark (DTU), Bureau Veritas (BV) and Nippon Kaiji Kyokai (ClassNK).

The third author acknowledges the support of the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MEST) through GCRC-SOP (Grant No. 2011-0030013).

REFERENCES

- Berberović, E., van Hinsberg, N.P., Jakirlić, S., Roisman, I.V. and Tropea, C., 2009. Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution. *Physical Review E*, 79, pp.036306-1.
- Bos, F.M., van Oudheusden, B.W. and Bijl, H., 2013. Radial basis function based mesh deformation applied to simulation of flow around flapping wings. *Computers & Fluids*, 79, pp.167-177.
- Cabos, C., Dilba, B., Krömer, M., and Schwenkenberg, A., 2011. Modal approach to fluid structure interaction applied to a ship in waves. In *Proceedings of the 3rd International Conference on Marine Structures, MARSTRUCT 2011*, Hamburg, Germany, 28-30 March 2011, pp.9-17.
- Carr, J.C., Beatson, R.K., Cherrie, J.B., Mitchell, T.J., Fright, W.R., McCallum, B.C. and Evans, T.R., 2001. Reconstruction and representation of 3D objects with radial basis functions. In *Proceedings of the 28th annual Conference on Computer graphics and interactive techniques (SIGGRAPH '01)*. New York, USA, 12-17 August 2001, pp.67-76.
- Causin, P., Gerbeau, J.F. and Nobile, F., 2005. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Computer Methods in Applied Mechanics and Engineering*, 197(42-44) pp.4506-4527.
- Farrell, P.E. and Maddison, J.R., 2011. Conservative interpolation between volume meshes by local Galerkin projection. *Computer Methods in Applied Mechanics and Engineering*, 200(1-4), pp.89-100.
- Hirt, C.W. and Nichols, B.D., 1981. Volume of Fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39, pp. 201-225.

- Hou, G., Wang, J. and Layton, A., 2012. Numerical Methods for Fluid-Structure Interaction - A review. *Communications in Computational Physics*, 12(2), pp.337-377.
- Irons, B. and Tuck, R.C., 1969. A version of the Aitken accelerator for computer implementation. *International Journal for Numerical Methods in Engineering*, 1, pp.275-277.
- Issa, R.I., 1985. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics*, 62, pp.40-65.
- Jacobsen, N.G., Fuhrman, D.R. and Fredsøe, J., 2012. A wave generation toolbox for the open-source CFD library: OpenFoam®. *International Journal for Numerical Methods in Fluids*, 70(9), pp.1073-1088.
- Jasak, H. and Tuković, Ž., 2006. Automatic mesh motion for the unstructured finite volume method. *Transaction of FAMENA*, 30(2), pp.1-20.
- Jensen, J.J., 2001. Load and global response of ships. *Elsevier Ocean Engineering Series*, 4, pp.1-337.
- Kim, Y., Kim, K.H. and Kim, Y., 2009. Time domain springing analysis on a floating barge under oblique wave. *Journal of Marine Science and Technology*, 14, pp.451-468.
- Malenica, Š., Molin, B., Remy, F. and Senjanović., 2003. Hydroelastic response of a barge to impulsive and non-impulsive wave loads. *In proc. Hydroelasticity in Marine Technology*, 3, pp.107-116.
- Muzaferija, S., Peric, M., Sames, P. and Schellin, T., 1999. A two-fluid Navier-Stokes solver to simulate water entry. *Twenty-Second Symposium on Naval Hydrodynamics*, Washington D.C., USA, 9-14 August 1998, pp.638-651.
- Oberhagemann, J., Shigunov, V. and el Moctar, O., 2012. Application of CFD in long-term extreme value analyses of wave loads. *Ship Technology Research Journal*, 59(3), pp.4-22.
- Piro, D.J. and Maki, K.J., 2013. Hydroelastic analysis of bodies that enter and exit water. *Journal of Fluids and Structures*, 37, pp. 134-150.
- Remy, F., Molin, B. and Ledoux, A., 2006. Experimental and numerical study of the wave response of a flexible barge. *4th international conference on hydroelasticity in marine technology*, Wuxi, China, 10-14 September 2006, pp.255-264.
- Rusche, H., 2002. *Computational fluid dynamics of dispersed two-phase flows at high phase fractions*. PhD thesis, Imperial College, London.
- Seng, S., Andersen, I.M.V. and Jensen, J.J., 2012. On the influence of hull girder flexibility on the wave induced bending. *In Proceedings of the 6th International Conference on Hydroelasticity in Marine Technology*, Tokyo, Japan, 19-21 September 2012, pp.341-353.
- Shabana, A.A., 2010. *Dynamics of multibody systems*. 3 edition. Cambridge: Cambridge University Press.
- Sherif, K., Irschik, H. and Witteveen, W., 2012. Transformation of arbitrary elastic mode shapes into pseudo-free-surface and rigid body modes for multibody dynamic systems. *Journal of Computational and Nonlinear Dynamics*. 7, pp. 021008-1.
- Ubbink, O. and Issa, R., 1999. A method for capturing sharp fluid interfaces on arbitrary meshes. *Journal of Computational Physics*, 153(1), pp.26-50.
- Youngs, D.L., 1982. Time-dependent multi-material flow with large fluid distortion. *Numerical Methods for Fluid Dynamics*, 24, pp.273-285.
- Wasfy, T.M. and Noor, A.K., 2003. Computational strategies for flexible multibody systems. *Applied Mechanics Reviews*, 56(6), pp.553-613.
- Weller, H.G., Tabor, G. and Jasak, H. and Fureby, C., 1998. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Journal of Computational Physics*, 12(6), pp.620-631.